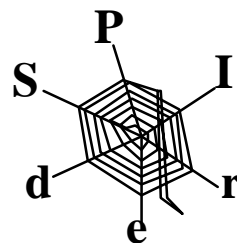


SPIDER Koerier



December 2006 Nummer 5 www.st-SPIDER.nl

n Redactioneel

Voor u ligt de 5^e en tevens laatste SPIDER Koerier van 2006. Het was een bewogen jaar op alle fronten: in het bestuur (3 wisselingen), de evenementen (Conferentie, plenaire sessies en Q-events), alsook de positieve trend in ons vakgebied.

We hopen dat u de artikelen weer met veel plezier en interesse zult lezen en wensen u hele fijne dagen en een goed begin van 2007!

Mocht u nog een mededeling, suggestie of een artikel hebben waarvan u denkt dat het interessant zou kunnen zijn voor de SPIDER leden, mail dan naar: koerier@st-SPIDER.nl.

n Inhoudsopgave

Redactioneel.....	0
Inhoudsopgave	0
Van het Bestuur	0
SPIDER op de ESEPG 2006	2
Managing Time Pressure in New Product Development teams	2
Build-architectuur krijgt te weinig aandacht.....	5
Requirements and Acceptance Management (part 3).....	8
Infotenties.....	14
Ben als SEI affiliate.....	14
Verslag workshop roadmaps	16
Verslag Q sessie.....	16
De werkgroepen.....	18
Nieuwsberichten & evenementenkalender ...	19
Deelname in SPIDER	19
Colofon.....	19

cijfers gepubliceerd die dit bewijzen. Er is volop onderzoek gedaan, zowel naar "oudere" technieken zoals inspecties, maar ook naar *agile* technieken zoals *pair programming* en *test driven design*. Helaas dringen de cijfers en onderzoeksresultaten maar mondjesmaat door, en dat is jammer. Anno 2006 kan verbeteren niet meer zonder een business case, en dit soort feiten dienen het management overtuigen dat verbeteren van de kwaliteit zich terugbetaald in geld en tijd.

SPIDER ziet het als een van haar taken om de brug te slaan tussen onderzoek en ontwikkeling. In onze plenaire sessies en de conferentie programmeren we presentaties waarin resultaten uit onderzoek getoond worden. We publiceren artikelen in de koerier over dit onderwerp. In diverse werkgroepen worden resultaten uit de onderzoekswereld toepasbaar gemaakt, bv. door boekjes met een overzicht van de beschikbare kennis en starterskits. We promoten conferenties en onderzoek met als doel om de industrie erbij te betrekken. In al deze activiteiten geven we extra aandacht aan cijfers en feiten, omdat we van mening zijn dat het essentiële informatie is voor implementatie van verbeterprogramma's.

Know your figures geldt uiteraard ook voor onszelf. Even op een rijtje: In 2006 hebben we 7 evenementen georganiseerd, 3 plenaire sessies, 2 conferenties (SPIDER jaarconferentie en Society for ICT Quality) en 2 workshops (CMMI roadmaps en Kwaliteit in ICT, DAT kan IK een stuk beter!). Ook hebben we aan de lancering van de kleine CMMI meegewerkt, en aan de ESEPG conferentie. De conferentie was met bijna 100 bezoekers een succes. Er zijn 5 koeriers uitgebracht

n Van het Bestuur

2006 zit er weer bijna op! Het gaat beter in de ICT, en dat zie je ook in procesverbetering en kwaliteitszorg. Er is weer ruimte om te verbeteren, sterker nog, het moet! Hogere productiviteit en kortere ontwikkeltijden zijn noodzakelijk. En dan ontstaan er discussies over de rol van kwaliteit. Kan het goedkoper, sneller, en beter? Of leidt de hoge druk op tijd en kosten tot slechtere kwaliteit? Het lijkt allemaal niet eenvoudig...

De feiten zijn er. We weten allemaal dat het voorkomen van fouten, en eerder vinden van fouten veel goedkoper is (en sneller) dan oplossen nadat het product bij de klant staat. Er zijn voldoende

De activiteiten van SPIDER worden gesponsord door financiële bijdragen van:



Philips.com

Kza.nl

Sogeti.nl

Sipartners.nl

Pstestware.com

gedurende het jaar, en met zo'n 20 mailings hebben we u op de hoogte gehouden van belangrijke events op het gebied van SPI en QA. Het aantal leden, donateurs en sponsors is gegroeid met gemiddeld 20%. Het bestuur bestaat weer uit 5 personen, we zoeken nog een 6e persoon. Interesse? Neem contact met mij op. Financieel staan we er goed voor met voldoende middelen om in 2007, waarin we 10 jaar bestaan, goed uit te pakken.

Wij gaan door op onze missie van verbetering van software ontwikkeling en kwaliteit in 2007. Gaat u met ons mee? We zien u graag bij een van onze evenementen! Ik wens u een gezond en een kwalitatief en productief 2007!

Namens het bestuur,
Ben Linders, voorzitter.

n SPIder op de ESEPG 2006

ESEPG 2006: SPIder workshop over "SPI results, trends & future challenges"

Op de European Software Engineering Process Group Conferentie (ESEPG) heeft SPIder een workshop georganiseerd over "*SPI results, trends & future challenges*". In een groepsdiscussie werd aan de hand van een aantal stellingen bediscussieerd welke uitdagingen spelen bij de implementatie van SPI. In dit artikel staan de belangrijkste conclusies uit de workshop.

De discussie werd gestart met een stelling:

- SPI has been deployed for many years.

One would expect that with all that experience:

- the software industry would be mature by now;
- a disciplined engineering based approach;
- using scientifically proved methods;
- resulting efficient and effective processes;
- leading to controlled development of products;
- steered by quantitative investment decisions.

Unfortunately, that is not where we are...

Deze stelling ontlokte (uiteraard!) de nodige discussie. Daarin werd gezocht naar de belangrijkste oorzaken waarom SPI nog onvoldoende volwassen is. De volgende "*challenges*" zijn geformuleerd:

- SW development is an engineering, people should be trained for 4-5 years
- It is an design discipline, not manufacturing; treat it appropriately
- There is no build up body of [shared / available / accessible] knowledge
- We are still re-inventing i.s.o. using processes available/"of the shelf"

Aanpakken van bovenstaande conclusies is volgens de workshop deelnemers essentieel om SPI tot een succes te maken. De 1e bullit heeft betrekking op de curriculums van software engineering opleidingen. Een aantal hogescholen en universiteiten werkt heel zeer actief aan, deze

verandering is dus wel ingezet maar het zal nog jaren duren voordat dit effect merkbaar is in de industrie. Over de 2e bullit: SPIder stimuleert het denken over processen vanuit een systems engineering perspectief, en zoekt daarin naar oplossingen hoe zulke processen effectief gemanaged kunnen worden. De 3e bullit is iets waar SPIder een belangrijke rol speelt, middels het netwerk en evenementen wordt kennis zichtbaar gemaakt en gedeeld in de Nederlandse SPI gemeenschap. Bullit 4 is iets waar SPIder ook naar kijkt, maar wat moeilijk te veranderen is omdat het zich grotendeels binnen organisaties afspeelt, desondanks denken we dat het delen van kennis en ervaring over processen de kans op hergebruik zal vergroten. Samenvattend: De workshop laat de uitdagingen voor de SPI zien; SPIder gaat deze uitdagingen aan.

Ter info: De volledig presentatie van de workshop staat op de SPIder website: <http://www.spider.nl/doc.php?Doc=125&ID=e590f311ac9202c98dedca47dc50b99d>

Cees Michiels

n Managing Time Pressure in New Product Development teams

• Introduction

Time pressure is a common experience of teams developing new products. However, there are limited empirical studies that are conducted to understand how perceived time pressure influences team processes and team performance. The aim of the article is multiple. First, I will provide you with theories that have evolved in the last decades to describe the effects of time pressure on individuals and teams. Second, I will present to you results of my findings on 'Managing time pressure in new product development (NPD) teams' in hope that it will give you ideas on how you can better navigate your teams during time pressure periods. Finally, I will conclude this article with what's next for my research, and what you can expect from my work in the next twelve months.

• The Attentional Focus Model, and Need for Cognitive Closure

In this Section, I introduce the Attentional Focus Model, and Need for Cognitive Closure to provide you with insights that are grounded in theories on how time pressure affects individuals.

The Attentional Focus Model: Karau and Kelly [1] introduced the Attentional Focus Model (AFM) to explain the intricate relationship between time pressure and performance. The basic premise behind AFM is that deadlines restrain time resources of individuals or teams, which in turn affect the content and the amount of information that an individual or a team attends to in its task environment. The AFM suggests that time pressure serves to narrow team members' cognitive resource

and focus, which cause features that appear most central to achieving a task to increase in relative salience, and features that seem less beneficial to task completion to become less important. A central implication of the AFM is that time pressure can either enhance or reduce performance depending on the requirements for successful task completion. If time pressure leads a team to focus on optimal amount of information, then performance should improve as a result of time pressure. However, in most cases, especially in new product development, tasks are relatively complex. They require extensive collaboration, and in-depth and repetitive information exchanges between members, teams and departments prior to arriving at an optimal outcome. Under such scenarios, time pressure plays a less facilitative role. In fact, time pressure is likely to restrict either the amount or the thoroughness with which information is shared, thereby reducing performance.

The AFM is consistent with a number of theories stating that stress lead individuals to attend to an increasingly narrow range of cues [2], and that processing objectives influence the content of the group interaction, which in turn influence group performance. Hence, the AFM is grounded in a long tradition of cue utilization and group process-group performance theories [3].

Need for Cognitive Closure: Kruglanski [4] coined the term Need for Cognitive Closure (NCC) to explain the closing of the cognitive mind under increasing pressure to achieve a goal. A comparison between AFM and NCC reveals that the former concentrates on cognition, while the latter uses both the cognitive and the motivational (affective) aspects of individuals to explain time pressure. NCC captures the fact that people stop considering multiple alternatives, engage in shallow rather than thorough information processing, and refrain from critical probing of a given seemingly adequate solution or judgment (Kruglanski & Freund, 1983). NCC occurs when the pressure is so high that it diminishes the ability and motivation to perform the task thoroughly. For example, need for closure is heightened when the apparent benefit of closure is the ability to decide in time for an important deadline. Thus, individuals under high time pressure experience a high need for closure. Early closure removes the necessity for further information processing, which can be both aversive and frustrating [5].

The theories showed that individuals narrow their focus on the important tasks or cues, and are inclined towards cognitive closure when they experience levels of time pressure that are beyond their optimal thresholds. Hence, individuals function most effectively when the experienced pressure matches their optimal thresholds [6], and high levels of time pressure are likely to cripple the cognitive performance of individuals. Hence, both theories support that perceived time pressure has an inverted-U relationship with performance.

The curvilinear relationship between time pressure and performance shows that time pressure has its positive impact on team performance if time pressure is well harnessed – like a sail that is wisely manoeuvred to capture the wind propels its boat

most effectively onward. Time pressure is the circumstance and consequence of the environment in which product development teams operate in. Instead of submitting to the fact that the experience of time pressure is ubiquitous and nothing else can be done to improve the environment, my work goes on to investigate the contextual factors that enable individuals and teams to cope with the time pressure at the workplace. These factors help management to better manage its teams during high time pressure periods.

- Contextual factors that help NPD teams to cope with time pressure

In this section, results were based on in-depth interviews (with Critical Incident Techniques) with 26 project managers, project leaders, and developers from Germany and The Netherlands. Results show that the contextual factors can be categorized into (1) Management Intervention, (2) Team Characteristics, and (3) External Elements [7].

1. Management Intervention: Results showed that management play a vital role in managing time pressure in teams. A culture for open vertical communication is needed for management intervention to be effective. In many occasions, the informants in this study took the initiative to feedback to the higher management when high time pressure was crippling their effectiveness. Subsequently, time pressure was reduced when the management acted upon the information and came up with workable solutions.

- Goal Significance:** Goals keep individuals focused on meeting one or more targets, and cause members to feel personally challenged. Nonetheless, having goals alone is insufficient. They need to be communicated, understood, and perceived to be important. Projects with well-conversed goals, in general, help their team members to understand the reasons for going through the time pressure period.

- Participative Scheduling:** The felt autonomy, which is provided to the developers when asked to be involved in schedule planning allowed the developers to give a realistic time estimation of the tasks, and helped them to be committed to the goals.

- Shielding:** Disturbances steal time and take concentration away from members. Disturbances may emanate from outside (requests that take the developers' time from the real tasks) and/or inside the teams (requests related to the project but are not directly related to the immediate task that a developer is working on). Therefore, actions or people that reduce these disturbances actually help teams in times of time pressure.

- Prioritization:** Prioritization is a special type of instrumental support (see below), and is a decision that can largely be made by the project leaders or managers. In general, team members require undivided concentration when solving complex problems. Having multiple tasks of equal priorities in mind does not help members to focus and is likely to cause members to perceive more

time pressure. Here, clearly prioritizing the team's tasks is an event that helps the team to cope with time pressure.

2. Team Characteristics: Many organizations have adopted the team-approach to new product development for pragmatic reasons for example to generate innovative ideas, to share knowledge, and to solve complicated problems. Results showed that team-approach provides the social environment for team members to cope with high time pressure. This section highlights to the management the importance of nurturing a NPD team that has the following characteristics:

- Team Commitment: An environment where all the key members of a team work hard and display high commitment to reach the team goals spurs all individuals to exert high levels of dedication and effort to accomplish their personal goals.
- Team Confidence: A confident team believes in their abilities and is less crippled by uncertainties. Such a team also perceives time pressure as a stimulant rather than a threat.
- Team Support:
 - o Informational Support: In face of a development problem, members typically had to spend a large amount of time troubleshooting and acquiring the appropriate knowledge to solve the problem. The time for problem solving can be shortened and time pressure reduced, if knowledge and past experiences are openly shared among team members.
 - o Emotional Support: An advantage of working in teams is the social relationships that it provides to the members. The opportunity for open expression of feelings and contact with others in team is helpful during stressful times.
 - o Appraisal Support: Team members tended to cope with time pressure by working longer hours. People wanted to be appreciated and rightly recognized for the results that they deliver, especially after working overtime.
 - o Instrumental Support: Having too much to do in too little time is a common experience for individuals in the NPD teams. They generally experienced cognitive resource constraints and thus perceived high levels of time pressure. Therefore, proactive actions taken by the team members or by the higher management that help to reallocate tasks and resources is a practical way that helps teams to cope with time pressure.

3. External Elements are factors from outside of the organization that influence how team members perceive time pressure. Though, these are factors that are not well within the control of management, they should nonetheless be made aware of. The results revealed Customer Involvement, and Family Support as the two external elements.

The experience of time pressure is going to stay for NPD teams. The aim of the results is not to eliminate time pressure for it has positive and stimulating effects on team performance, and it is wishful thinking to reduce time pressure to ground-zero. Market-driven time pressure is part of the NPD work environment. The goal of the results is to help managers to be more sensitive to the contextual

factors that are particularly useful in reducing the organization-induced time pressure (i.e. lack of prioritization), and to engage in practices that enable their team members to perceive time pressure positively.

• What's Next?

I have been investigating the impact of time pressure on teams in the NPD environment based on the input-process-output model. Perceived time pressure being the input, how does it affect the team processes (i.e. communication) and team performance (i.e quality)? In addition, I included moderators in the model with the aim of confirming the contextual factors that significantly improve the input-process relationship. The model will be tested using questionnaires and statistical analysis. Two questions will be answered at the end of the investigation: (1) What are the positive and negative impact of time pressure on NPD team processes, and NPD team performance?, and (2) What are the team environments that moderates the negative effects that time pressure have on team processes. I will share with you more of the results in time.

I am currently looking for NPD teams to take part in a short one-time online survey. If you are interested in the research and how you can contribute to the results, please feel free to contact me at d.s.f.chong@tm.tue.nl

NOTE: Results in Section II were published in the Proceedings of the PDMA Research Conference 2005, "Managing Time Pressure in New Product Development", San Diego, California, USA, Oct 22-23, 2005, pp. 248-252

References:

- [1] Karau S. J. & Kelly J. R. (1992). The effects of time scarcity and time abundance on group performance quality and interaction process. *Journal of Experimental Social Psychology*, 28: 542-571.
- [2] Kruglanski, A.W. and Webster, D. M. (1996). Motivated closing of the mind: "Seizing" and "Freezing". *Psychological Review*, 103(2): 263-283.
- [3] Karau S. J. & Kelly J. R. (2004). Time pressure and team performance: An attentional focus integration. *Time in groups*, 6: 185-212.
- [4] Kruglanski, A. W. (1989). *Lay Epistemics and Human Knowledge: Cognitive and Motivational Bases*. New York: Plenum.
- [5] Kruglanski, A.W. & Freund, T. (1983). The freezing and un-freezing of lay inferences: Effects on impression primacy, ethnic stereotyping and numerical anchoring. *Journal of Experimental Social Psychology*, 19: 448-468.
- [6] Andrew, F. M. and Farris, G. F. (1972). Time pressure and performance of scientists and engineers: A five-year panel study, *Organizational Behavior and Human Performance*, 8: 185-200.
- [7] Chong D.S.F.,W v Eerde, C.G. Rutte, K.H. Chai and A.C. Brombacher, Managing Time Pressure in New Product Development, Proceedings of the PDMA Research Forum 2005, San Diego, California, USA, Oct 22-23, 2005, pp. 248-252



Darrel Chong is a Ph.D candidate of National University of Singapore and Eindhoven University of Technology. His research focuses on time pressure, new product development teams, and team communication.

E-mail: d.s.f.chong@tm.tue.nl

n Build-architectuur krijgt te weinig aandacht

• Inleiding

Het aandeel van software in systemen is groot en zal in de toekomst alleen nog maar toenemen. Tevens neemt niet alleen de complexiteit van deze software intensieve systemen toe maar ook de snelheid waarmee wijzigingen doorgevoerd moeten worden. In de praktijk lopen bedrijven hier vaak tegen problemen op: om veranderingen uit concurrentie overweging snel te kunnen doorvoeren ontstaat architectuur erosie. Hierdoor wordt het voor de configuration manager steeds lastiger om niet alleen software maar ook de wijzigingen op een consistente manier te beheersen.

Interface wijzigingen in het product en daarmee wijzigingen in de structuur van opslag in het software configuration management systeem gaan hand in hand met (toenemende) lange build tijden van de software, uit de hand gelopen archief structuren en build afhankelijkheden. Oplossingen worden gezocht in het uitbreiden van het machinepark met state-of-the-art systemen en tools. Dit biedt echter maar korte tijd soelaas. Op den duur ziet men het software configuration management systeem en daarmee configuration management en build management als de beperkende factor voor het feitelijke product creatie proces. Maar is dat wel terecht?

In dit artikel gaan we op zoek naar de oorzaken en mogelijke oplossingen om lange buildtijden te voorkomen.

• Top

De ervaring leert dat menselijke factoren een niet te onderschatten rol spelen in softwareontwikkeling – tenslotte zijn het nog altijd mensen die software maken. Het TOP-model (Techniek, Organisatie en Proces) integreert deze observatie in een samenhangende, pragmatische visie op softwareontwikkeling. Bij dit model staan technische, organisatorische en procesmatige factoren met elkaar in verband, om in die onderlinge

samenhang het succes van een (software) ontwikkeltraject te bepalen.

Elke beslissingen moet worden genomen met inachtneming van de balans tussen deze drie factoren. Vaak werkt juist de afwezigheid van de synergie tussen techniek, organisatie en proces als zand tussen de tandwielen.

• Kritische succesfactoren en tips voor een succesvol configuration management systeem

Techniek:

- state-of-the-art machine park en tools,
- Interconnecties tussen modules als drijfveer gebruiken voor de C.M. archief structuur

Organisatie:

- levering van nieuwe software onder hoge druk, snelle wijzigingen
- noodscenario's zijn bekend, maar worden alleen in echte noodsituaties (escalaties) gebruikt

Proces:

- wijzigingsprocedure is toereikend wordt gevolgd.
- wijzigingsprocedure voorziet in een escalatie mechanisme.

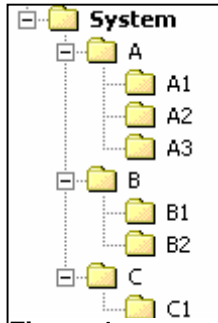
• Software Architectuur en Configuration Management

Een software architectuur wordt vaak beschreven vanuit verschillende views (gezichtspunten). De gekozen set van views is afhankelijk van het domein, maar kent wel een aantal steeds terugkerende onderdelen. Bijvoorbeeld volgens het Soni-Nord-Hofmeister – model bestaat een architectuurbeschrijving uit een conceptuele view, een module inter-connectie view, executie- en een code view. Van belang voor CM is de module inter-connectie view die aangeeft hoe de architect de decompositie van het systeem ziet en de code view, datgene wat de ontwikkelaar ziet op het filesysteem.

De code view is uiterst belangrijk voor de configuration manager. Het configuration management systeem moet er namelijk voor zorgen dat de gewenste configuration items in de juiste structuur aanwezig zijn, zodanig dat de architectuur gewaarborgd blijft en dat de uiteindelijke deliverable(s) op een correcte, reproduceerbare en traceerbare wijze worden opgeleverd. Hier wordt vaak – bijna onbewust – gekozen voor de module inter-connectie view als basis voor de code view. Met andere woorden: modules, sub-modules en componenten met hun hiërarchie worden 1-op-1 afgebeeld op het configuration management systeem en daarmee ook op het filesysteem. Deze keuze wordt veelal gemaakt omdat het idee bestaat dat 'het dan gemakkelijker is voor ontwikkelaars'.

Zoals we zullen zien, kan de praktijk anders uitwijzen.

Een voorbeeld uit de dagelijkse praktijk (zie hieronder)



Figuur 1

Een bepaald project heeft een archiefstructuur gekozen en is hier sinds enige tijd mee aan de slag. Ter info: A3 en B2 worden ingekocht en moeten regelmatig, tegelijkertijd, worden vervangen door nieuwe versies. A1, A2, B1 en C1 worden binnen het project ontwikkeld. Waarbij B1 afhankelijk is van C1.

De praktijk:

Zodra een nieuwe levering van A3 en B2 geïntegreerd dient te worden ontstaan er problemen. Het overstappen naar een nieuwe levering wordt gedaan door minimaal 2 ontwikkelaars die hier onafhankelijk van elkaar opdracht toe krijgen. Weliswaar gebruikt het project een continu integratieproces, maar elke keer gaat het mis. De integratie mislukt. En er is een aantal correctieve acties nodig. Tijdens deze integratie stappen staat het project als het ware stil omdat de voortgang nihil is.

De oorzaken:

Het proces om een nieuwe levering in het archief te zetten is niet optimaal afgestemd op de gegeven situatie. Wat betreft de organisatie zien we dat de archiefstructuur in combinatie met het gebruikte configuratie management systeem van invloed te zijn op deze integratie.

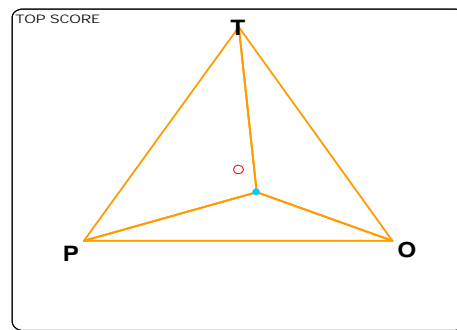
De techniek die gebruikt wordt om het software systeem te bouwen is niet optimaal omdat met name afhankelijkheden tussen diverse modules niet of slechts gedeeltelijk worden bekeken.

Bovenstaande situatie bekeken vanuit het TOP model laat een duidelijke onbalans zien tussen de 3 assen T-, O- en P-as, hetgeen het bewijs is dat het configuration management proces niet soepel verloopt. Herstellen van de balans is alleen mogelijk door op minimaal 2 assen verbeteracties te definiëren en uit te voeren. Uitsluitend aanpassen van een van de assen zal dus niet de gewenste balans opleveren.

De doorgevoerde verbeteringen vanuit TOP gedefinieerd:

Techniek:

- aangepast buildproces dat meer modulair is opgezet met duidelijk vooraf gedefinieerde include-hierarchy.
- de archiefstructuur van het project aangepast waarbij het nu vanuit het configuration management systeem mogelijk is om een nieuwe levering van A3 en B2 tezamen als baseline van BL2 te gebruiken. A1 en A2 werden verschoven onder BL1. C1 en B1 die een sterke relatie hebben worden onder BL3 gebracht.



Figuur 2

Processen:

- ander integratie proces van A3 en B2 zodanig dat deze altijd tegelijkertijd worden geïntegreerd. Ook de verantwoordelijkheden voor de feitelijke integratie werd aangepast. De ontwikkelaars worden middels het configuration management systeem gedwongen om beter samen te werken.

Resultaat: het project kan nu eenvoudiger omschakelen naar een nieuwe levering van A3 en B2. Het feitelijke integratieproces verloopt ook vele malen soepeler.

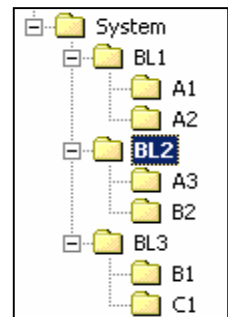
Ook de problemen tijdens het bouwen van de software zijn afgenomen doordat het nieuwe build proces goed rekening houdt met de afhankelijkheden tussen de diverse modules.

Laten we de 3 assen van het TOP-model eens langs lopen om te zien wat er in detail gebeurt.

• Techniek

Een gevolg van de keuze om de module inter-connectie view 1-op-1 af te beelden op de code view is dat elke wijziging in de inter-connectie view direct zijn invloed heeft op de code view. Omgekeerd kan ook een code view wijziging direct invloed hebben op de inter-connectie view.

Wanneer module A1 in figuur 1 direct wil communiceren met module B1 (hetgeen niet is toegestaan volgens de architectuur) kan dit eenvoudig door bijvoorbeeld het "include path" voor de compiler in de makefile aan te passen, of door een symbolische link (Unix) toe te voegen waardoor B1 toch "binnen bereik" van A1 komt. Dit heeft als resultaat dat de relatie tussen de module inter-connectie view en de code view niet meer 1-op-1 is. Iets waarvan tot op dat moment toch steeds vanuit werd gegaan. Langzaam maar zeker raken de module inter-connectie view en de code view verder van elkaar verwijderd, terwijl deze juist keurig gesynchroniseerd zouden moeten zijn. Intussen zie je dat de tijd die nodig is om het product te compileren en linken toeneemt en toeneemt.



Figuur 3

Het gevolg van deze toename is dat allereerst de ontwikkelaars gaan klagen over de algemene performance van het configuration management systeem. Zij zullen hierin worden gesteund door het

feit dat het genereren van de eigenlijke software aantoonbaar meer en meer tijd kost. Vervolgens gaat ook de organisatie in z'n geheel meer en meer klagen. De oplossing voor dit symptoom wordt gezocht in de technisch hoek. Aanschaf van een state-of-the-art machine park inclusief bijpassende tools moet de negatieve spiraal doorbreken. Dit geeft gedurende korte tijd weliswaar wat meer lucht, maar na verloop van tijd – en dat is sneller dan verwacht – komt men weer in de oude situatie terug. Niet alleen de ontwikkelaars maar ook het hoger management raakt hierdoor meer gefrustreerd.

• Organisatie

Al bij start van het project beginnen krachten te werken op het configuration management systeem en daarmee op de structuur van configuration items in de vorm van architectuurwijzigingen. Requirements die niet voorzien waren moeten op het allerlaatste moment worden geïmplementeerd. Showstoppers (bugs) komen op het allerlaatste moment aan de oppervlakte en moeten snel worden opgelost. Vanuit de organisatie is de druk hoog om de aanstaande levering – met de nieuwe requirements geïmplementeerd – doorgang te laten vinden, want anders gaat de klant weg of zijn er zware boete clausules. De verleiding is groot om te kiezen voor een snelle korte termijn oplossing (of beter: symptoombestrijding) die niet in lijn is met de module inter-connectie view. Bijvoorbeeld door middel van het aanroepen van een functie van een module (module B1 in figuur 1) die niet in zicht is (vanuit module A1 in figuur 1).

Doorvoeren van wijzigingen aan het product is weliswaar keurig beschreven in het wijzigingsproces, de druk vanuit de organisatie is echter zodanig groot dat men zich hieraan niet kan/wil houden. Desondanks wordt plechtig de belofte gedaan dat de wijziging later op een nette manier wordt geïmplementeerd. En daar blijft het bij. De architecturale views worden niet aangepast omdat het wijzigingsproces voor het gemak vergeten wordt. Dit geldt overigens ook voor de documenten die hieraan gelieerd zijn.

Dat dit niet een eenmalige actie is moge duidelijk zijn. Het gevolg van deze werkwijze is dat de architectuur aan 'erosie' onderhevig is. De architect staat daardoor regelmatig voor voldongen feiten: de procedure voor wijzigingen wordt nauwelijks of niet gevolgd. Door het niet volgen van de procedures komt er dus ook geen formele terugkoppeling naar de architect om de architectuur beschrijvingen aan te passen: blijkbaar is hij op dat moment niet meer verantwoordelijk en bevoegd. Het aantal hacks waarbij zelfs de architect niet eens meer betrokken wordt neemt explosief toe. De genoemde architectuur views (module inter-connectie view en de code view) komen steeds verder van de werkelijkheid af te staan. Het TOP model raakt uit balans door de hoge druk vanuit de organisatie.

• Proces

We zien dat het bestaande proces wordt losgelaten want het wijzigen van de architectuur, code en codestructuur is niet goed onder controle. De wijzigingsprocedure zelf staat hier absoluut ook ter discussie, er vanuit gaande dat deze procedure

bestaat. Men is zich helaas niet altijd bewust dat het zuiver en alleen beschrijven van dit proces nog geen adequate oplossing is. Naleving en controle is eveneens van groot belang. De kracht van deze procedure ontstaat juist wanneer hier ook een verantwoordelijke voor is die overeenkomstige verantwoordelijkheden en bevoegdheden heeft alsmede middelen om dit af te dwingen. In de praktijk zien we dat deze functie niet of slechts beperkt aanwezig is. Men kiest er voor om de symptomen te bestrijden door middel van (extra) controles gedurende het build proces. Dit heeft als resultaat dat het feitelijke build proces nog meer vertraging oploopt.

Daarnaast zien we dat ontwikkelaars op zeer korte termijn wijzigingen doorvoeren die een software probleem als zodanig weliswaar oplossen maar dat ze de tijd niet nemen/krijgen om tot een gedegen oplossing te komen die in overeenstemming is met de architectuur, én met de wijzigingsprocedure. Verwarring bij de ontwikkelaars neemt meer en meer toe omdat:

- De architecturale view niet meer in overeenstemming is met de code view
- Architectuur documentatie niet meer in sync is met het project
- Modules aangepast worden zonder dat de oorspronkelijke auteur daarvan op de hoogte is
- Documentatie van modules niet meer up-to-date is
- Nieuwe (onverwachte) en ongedocumenteerde build afhankelijkheden zijn ontstaan.

Het is echter niet alleen het proces van wijzigen dat hier belangrijk is. Verantwoordelijkheid voor de functionele inhoud van het product (welke feature(s) moet erin zitten, welke bugfixes moeten er wel en niet meegenomen worden) ligt vaak bij een integratie manager. Deze is er verantwoordelijk voor om de inhoud (en daarmee de kwaliteit) van het eindproduct te waarborgen en te verhogen, daarbij rekening houdend met de (on)mogelijkheden van het configuration management systeem. Deze rol kan namelijk de stuwende factor zijn om bepaalde bugfixes juist wel en andere juist niet of op een later tijdstip te integreren. Bij afwezigheid van deze rol (inclusief daaraan gekoppelde bevoegdheden) wordt de configuration manager meer en meer gefrustreerd omdat hij deze rol opgedrongen krijgt. Dit heeft onder meer tot gevolg dat het build proces minder frequent tot een goed einde komt.

• Ontkoppeling

De configuration manager moet zich bewust zijn van de risico's wanneer de module inter-connectie view 1-op-1 als code view wordt neergezet. Ontkoppeling van deze 2 domeinen volgens een goed gedefinieerde vertaling zal hem daarbij helpen (bijvoorbeeld door uit te gaan van de inter-connecties en niet van module hiërarchie in combinatie met een duidelijke code-eigenaarschap). Dit in nauwe samenwerking met de software architect en andere stakeholders.

Ook het handhaven van de wijzigingsprocedure die tot in detail beschrijft wat er gedaan moet worden in bepaalde 'nood' situaties (dus inclusief escalatie

mechanisme) zal bij alle betrokkenen aan het project (de organisatie) meer duidelijkheid bieden in de daaraan gekoppelde processen. Dit geheel geeft op zijn beurt een positieve impuls aan de kwaliteit van het eindproduct én de organisatie.

• Conclusie

Vanuit de technische kant gezien is het logisch om ervoor te zorgen dat er voldoende aandacht wordt besteedt aan machinepark en tools. Voorkomen echter hierin door te schieten door de organisatorische en procesmatige aspecten te negeren.

Voor wat betreft de organisatie van de configuration items (code architectuur, code structuur) moet men goed nadenken over de fysieke samenstelling van de code structuur, met name door de architecten en de configuration manager.

Ontkoppeling van de code view van de module inter-connectie view is een prima aanzet, mits doorgevoerd in de configuration management structuur. Problemen in de module inter-connectie view leiden dan namelijk niet tot problemen in de code view en omgekeerd. Dit is al een grote stap voorwaarts. Mogelijke architecturale wijzigingen kunnen dan zonder problemen worden doorgevoerd.

Wees ervan bewust dat procesmatige en organisatorische aspecten veel aandacht nodig hebben (wellicht meer dan u denkt). Alleen op deze manier kunnen we beter omgaan met de snelheid waarmee tegenwoordig software intensieve systemen gewijzigd worden.

Auteurs: Frank van der Kruijsen

(frank.van.der.kruijsen@ict.nl)

Paul Moors (paul.moors@ict.nl)

n Requirements and Acceptance Management (part 3)

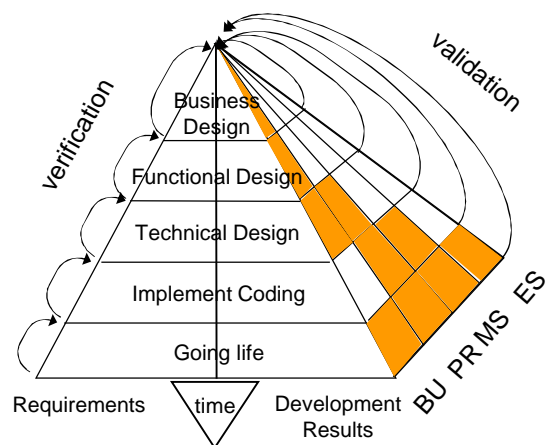
• Inleiding

In dit derde artikel in de serie over "Requirements and Acceptance Management" worden Best Practices behandeld en wordt ingegaan op bekende en minder bekende standaards voor procesverbetering en Requirements zoals kunnen worden gebruikt om de kwaliteit van Requirements, en daarmee de acceptatie van het eindresultaat voor de Business, te verhogen.

In deel 1 van deze serie werd ingegaan op de belangrijke rol die de diverse Stakeholders in een project spelen en dat deze Stakeholders tijdig worden geïdentificeerd zodat hun eisen en verwachtingen vroegtijdig bekend zijn [[zie referentie 1](#)]. Standpunt hierbij is dat alle Requirements van alle Stakeholders vooraf bekend moeten en hun rol en positie in de organisatie ('machtsbasis'), om bij uiteindelijke oplevering van het projectresultaat, de acceptatie niet alleen mogelijk te maken maar transparant en beheersbaar te houden. Onderstaand generiek Stakeholder model geeft dit nog eens weer.



In deel 2 van deze serie werd ingegaan op het trade-off proces, het proces waarin Stakeholders 'onderhandelen' over in projecten te realiseren functionaliteit op te nemen in systemen of releases van systemen [[zie referentie 2](#)]. In dat artikel werd ook aangegeven dat er naast 'gewone' Requirements ook 'negatieve' Requirements bestaan; dit zijn Requirements die een beperking opleggen aan de Requirements van de Business. Deze constraints staan als het ware aan de top van de 'Requirements piramide', zie onderstaand model.



Al eerder werd gesteld dat aan een professioneel Requirements Engineering & Management proces, eisen ten aanzien van de organisatorische, methodische, technische inrichting worden gesteld, plus eisen m.b.t. aspecten betreffende de 'human factor'. Denk bijvoorbeeld aan eerst een Proof of Concept uitwerken of een pilot project uitvoeren als de technologische oplossing onzeker is, of niet op grote schaal een project starten als de productspecificaties onduidelijk zijn of onvoldoende duidelijk is of er wel een voldoende afzetmarkt voor het product is.

Dit artikel gaat nader in op "**De voorkant van het automatiseren**" [[zie referentie 3](#)]. 'De mens of de organisatie die zijn behoefte formuleert aan een technisch hoogwaardig hulpmiddel, zijnde ICT. En die vanaf de eerste dag wordt geconfronteerd met een sterker wordende begripsverwarring en samenwerkingschaos, met onduidelijkheid over

doelen en gevolgen.'Zo begint het voorwoord uit dit enigszins gedateerd boek (1987), een boek dat een unieke, niet verkokerde visie op ICT, projecten en automatisering geeft. De werkelijkheid is niet in checklists en methodes te vangen menen de beide auteurs. De human factor in organisaties dient zijn eigen duidelijke plaats in te nemen. Dit artikel vertaalt hun toenmalig revolutionaire visie op de opzet van projecten in ICT en de rol van de gebruikers daarbij, naar de huidige wereld van Requirements en Acceptance Management. Hun visie op hun werkelijkheid weergegeven in '**de zeven schillen van de organisatie**' en '**de driegeleding**' blijkt nog even actueel en toepasbaar te zijn als destijds.

Informatiesystemen bestaan vaak al erg lang (vooral in de financiële wereld). Het ene IS vervangt hierbij vaak het andere. Als je nu aan de gebruiker van een IS vraagt hoe zijn business proces eigenlijk werkt, dan weet hij dat vaak niet. Kennis en intelligentie zit vaak verstopt in software. Beheerders willen die kennis vaak ook onvoldoende met andere delen (kennis maakt macht!) Zelfs het een op een vervangen van een bestaand systeem door een ander systeem op een technisch geavanceerd platform, is vaak een lastige operatie. In het ergste geval fungeert het oude systeem als norm voor het nieuwe systeem. Dat de functionaliteit één op één moet voldoen aan het bestaande systeem is ook een vorm van een requirement vindt men dan. Maar wat er als er het schaduwdraaien een verschil ontstaat tussen werking/resultaat van het nieuwe systeem versus het oude systeem, dan moet het project maar aantonen dat het oude systeem fout is?! Lastig dus. Dit pleit voor goede Requirements.

Voorheen was Requirements Analyse het domein van de informatieanalyse. De systeemanalist of informatieanalist werd verondersteld de vertaalslag te kunnen maken van organisatie vereisten (Business Needs) naar technische mogelijkheden. Via Vooronderzoek, Definitiestudie en dergelijke. De systeemanalist bestaat feitelijk niet meer. Dikke pakken papier met beschrijvingen van de huidige situatie, veranderingssituatie en beschrijving toekomstige situatie worden feitelijk niet meer gemaakt. En dat is misschien maar goed ook.

Als de klant niet weet wat hij wil of de provider niet weet of wat de klant wil wel realiseerbaar is, zijn bestuurlijke oplossingen denkbaar om het eindproduct/resultaat op te delen in verschillende stukken en elk stuk verschillend te behandelen. Dit hangt samen met de onzekerheid van het proces resp. van het product.

Wat voorbeelden van VRAAG en AANBOD in het kader van Requirements.

• Vraag (business)

- Klant stelt hoge eisen (geen 'zwarte T-Ford').
- Kwaliteit went snel (airco in je auto).
- Kwaliteit is subjectief, eenzelfde product wordt door de een vaak heel anders ervaren dan door de ander en in de tijd gezien slijt de waardering over het product snel. Kwaliteit went en verwent!

- Steeds korter wordende levenscyclus van producten en diensten en trendgevoeligheid.
- Veel distributiekkanalen voor klanten beschikbaar; markten zijn open voor vele mededingers en transparant.
- Voortdurende prijs erosie (consumer electronics).
- Positie in de markt. Ben je monopolist dan bepaal je de prijs en kan je je een bugje veroorloven. Zit je in een markt met vrije mededinging dan levert een foutje je snel verlies van marktaandeel en financieel verlies op. Denk bijvoorbeeld aan de huidige markt voor ziektekosten verzekeringen.

Ga daar maar eens specificaties voor opstellen!

• Aanbod (Service Suppliers)

- Concentratie en schaalgrootte.
- Mondialisering (verschillende talen en culturen maken het communiceren over Requirements niet gemakkelijker)
- Embedded software (afroepbaarheid is een probleem! Denk aan bugs in een raket of in een navigatiesysteem in auto's).
- Uitwijk van software ontwikkeling naar lage lonen landen (die nog wel eens minder productiviteit leveren dan we zouden willen!) .
- Bedrijven worstelen met Customer Intimacy aan de voorkant en met Operational Excellence aan de achterkant.
- Diploma inflatie! Roep om certificatie.
- (Internationale) concurrentie.

Wat is de praktijk? Vaak gaat de directie van een bedrijf naar het buitenland en laat zich daar een mooi business pakket aansmeren. Of de ICT afdeling dit maar even wilt invoeren in de organisatie en of die externe projectmanager dat proces maar even wilt managen!

• Behoeftte aan gezamenlijke taal

Er is in 'Requirements land' duidelijk behoefte aan een gezamenlijke 'taal'. De gebruiker moet de taal van de ICT-er kunnen verstaan en nog belangrijker de ICT-ers moeten zich in de gebruiker kunnen verplaatsen en zijn taal herkennen. Extra complicatie is dat er verschillende methoden en technieken voor Requirements Engineering en Management bestaan die nèt allemaal een iets afwijkende taal gebruiken. Deze toegepast in organisaties levert vaak weer een eigen invulling op. De Requirements Community hoopt dat internationaal aanvaarde modellen zoals CMMI [**zie referentie 4**] een einde zullen maken aan de voortdurende Babylonische spraakverwarring. Standaardisatie helpt beslist maar hier is méér voor nodig.

Het feit dat veel ontwikkelaars in een zeer technisch jargon communiceren, wat het proces van elkaar begrijpen verder bemoeilijkt, wordt ook vastgesteld in '**Metaforen in Requirements engineering**' [**zie referentie 5**]. Om de gezamenlijke beeldvorming te verbeteren en de communicatie te versoepelen worden metaforen gebruikt. Zo wordt het Internet wel eens vergeleken met een bibliotheek, terwijl dit in werkelijkheid twee totaal verschillende objecten

zijn; de vergelijking dient slechts om de communicatie erover te vergemakkelijken. De betrokken partijen moeten wel eenzelfde beeld hebben met de metafoor. Realiseer je dat vooral bij internationale samenwerking metaforen juist de oorzaak van miscommunicatie kunnen zijn!

Door een werkgroep van de Stichting SPIDER is een aanpak voor procesverbetering in de wat kleinere organisaties ontwikkeld '**De Starterkit voor procesverbetering bij kleine(re) IT-eenheden of organisaties**' [zie referentie 6] beschrijft het proces Opstellen Requirements resp. Beheren Requirements in duidelijke taal zonder wolligheid of jargon. Zo omvat het proces Opstellen Requirements het volgende:

- Bereid het opstellen van Requirements voor door inlezen en vragen bedenken
- Verzamel informatie bij de stakeholders, literatuur e.d.
- Orden de verzamelde gegevens
- Bespreek de resultaten met de stakeholders
- Vul het ontbrekende aan
- Maak concept requirements document
- Review intern
- Review met de klant
- Verwerk de review gegevens
- Verstuur definitieve requirements document
- Vraag om akkoord van de klant

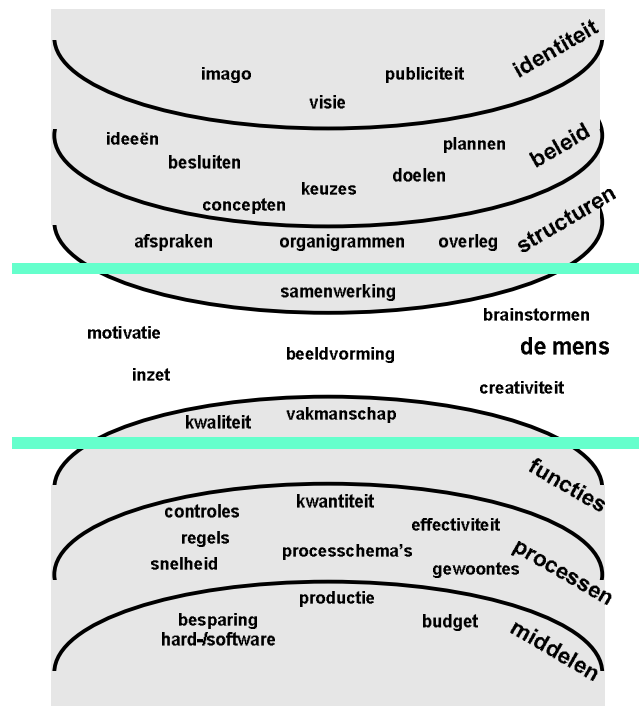
Jip en Janneke taal die iedereen kan begrijpen, een aanrader!

Dat het erg belangrijk is dat Requirements vooraf worden getoetst op technische realiseerbaarheid (bouwbaarheid) en testbaarheid wordt ook door Tom Gilb gezegd in zijn '**Requirements-Driven Management: A Planning Language**' [zie referentie 7]. 'We need to define Requirements in terms of desired future end states: testable and measurable'.

Natuurlijk is het inspecteren op bouwbaarheid en testbaarheid van groot belang, vooral als dit vroeg in de Ontwikkel Life Cycle wordt toegepast en men zich conformeert aan het resultaat, vergeet alleen niet dat alle Stakeholders betrokken moeten zijn. Er moeten derhalve ook vragen worden gesteld zoals: onderhoudbaarheid (de applicatie beheergroep), exploiteerbaarheid (de exploitant) en compliance met regulator Requirements (de Risk & Compliance afdeling), om er maar een paar te noemen.

• De 'zeven schillen van de organisatie'

Onderstaand model van Morssink en Kranendonk, De VOORKANT van het automatiseren [zie referentie 3] geeft een 'driegeleding' in drie subsystemen van de organisatie weer, elk met hun heel eigen belangen en werkwijze. Het gebied boven de (blauwe) streep is het politiek-strategische subsysteem, het gebied onder de streep het productie-economische subsysteem en ten slotte vormt het sociaal organisatorische subsysteem het middengebied.



Automatisering is van oudsher het sterkst vertegenwoordigd in de onderpool van de organisatie. Automatisering ontkent en verwaarloost het bestaan en het belang van het middengebied door de suggestie dat globale doelen in de bovenpool (visie, doelen, plannen) direct zijn te vertalen in een technische oplossing in de middenpool (naar processen en middelen). Automatisering van bedrijfsfuncties (bijv. door nieuwe diensten aan te kunnen bieden of nieuwe producten te kunnen voeren) vergt veel inzicht in de werkelijke behoeften van de klant en in de gewenste kwaliteit van producten en diensten. Deze behoeften raken alle schillen en niet alleen de onderste schil. Op het punt halverwege de functie schil houden echter de directe toepassingsmogelijkheden van automatisering op. Het hoogste dat automatisering kan is vervangen c.q. ondersteunen van het kwantitatief en kwalitatief beter afleveren van producten en diensten. Sommigen zijn van mening dat computers nooit kunnen worden ingezet om rechtstreeks te bereiken dat:

- mensen meer vakman worden
- mensen zich gelukkiger voelen
- mensen en groepen beter gaan samenwerken
- de organisatie beter functioneert
- het beleid helderder of doelmatiger is
- de bedrijfsidentiteit wordt opgevijseld, etc.

De fout die vaak door ICT-ers wordt begaan is om voorspelbare en beheersbare ICT processen met dito structuren, één op één geldig te verklaren in de hogere schillen van mens, organisatie en strategie. Het onjuiste en onvolledige beeld van de organisatie als mechanisme dus. Een nog grotere fout is als niet ICT-ers geloof hechten aan dit vernauwde

gezichtsveld van ICT-ers en hierin toch meegaan. Deze polarisatie levert toch niets op zul je zeggen? Klopt, maar het zorgt wel voor meer balans en minder ambitieuze verwachtingen in het bereiken van hoogstaande doelstellingen met de inzet van ICT. Er dient een voortdurende creatieve beweging plaats te vinden tussen de organisatie doelen en ideeën enerzijds en haar middelen en processen anderzijds. Op al deze fronten zijn Requirements van toepassing op dat wat door ICT gaat veranderen in organisaties. Wij missen domweg de creativiteit en het inzicht om hiervoor Requirements te definiëren. Over doelstellingen en samenhang tussen Organisatie & Cultuur, Methoden & Technieken, Hulpmiddelen/Tools en de Human Factor is al het nodige geschreven, zie Raamwerk Kwaliteit [zie referentie 8].

- Business Goals, Requirements en CMMI

Laten we dit integrale, holistische beeld van Morssink en Kranendonk eens loslaten op de modellen van CMMI en hier eens kritisch naar kijken. De bedenkers van CMMI zijn al net zo narrow focussed als de mensen die gebruik maken van de door hen ontwikkelde modellen. Zo worden de processen van software engineering nagenoeg één op één van toepassing verklaard voor andere engineering disciplines zoals mechanica en elektronica. De processen mogen weliswaar op elkaar lijken, wat men hiermee doet is echter volslagen anders. Dat wordt dan vergeten. Ook beginnen de CMMI Engineering processen voor het identificeren en vastleggen van Requirements niet of nauwelijks met het denken vanuit de Business Goals. Het wordt wel genoemd in CMMI maar niet geconcretiseerd. CMMI daalt namelijk erg snel (te snel) af naar Requirements die in de ICT wereld begrijpelijk zijn, de systeem of software Requirements in CMMI 'System Requirements allocated to software' genoemd. Nu is er niets tegen het gebruik van CMMI, in tegendeel, organisaties maken steeds meer en met succes gebruik van de concepten en Best Practices van procesinrichting en proces- en organisatieverbetering. Daar is niets mis mee tenzij de verwachting is dat 'de hele wereld in CMMI is te vatten en dat het opstellen van Requirements alleen maar een modelleringprobleem en beschrijvingsprobleem zou zijn. Niet dus! CMMI geeft een heldere definiëring van de verschillende soorten Stakeholders die gedecomposeerd in Engineering processen en subprocessen leiden tot steeds meerde gedetailleerde, heldere Requirements. Elke categorie van Requirements vormen de basis voor het opstellen van Designproducten m.b.t. die Requirements. Zowel Design producten als Requirements worden in de Ontwikkel Life Cycle verder gedetailleerd en concreet gemaakt opdat uiteindelijk het product of de dienst kan worden vervaardigd, geverifieerd, gevalideerd, geïntegreerd met andere componenten, geaccepteerd en voor

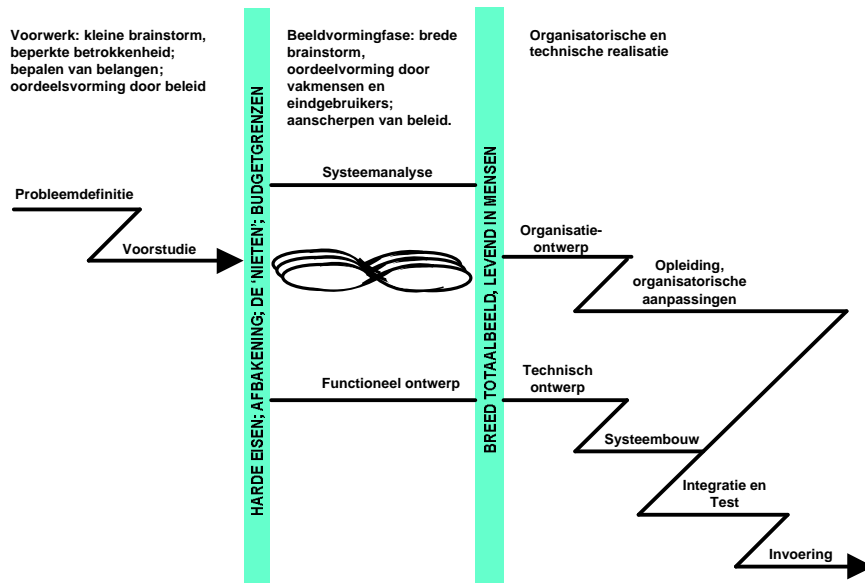
operationeel gebruik door de Stakeholders kan worden opgeleverd.

Neem nu de situatie dat eindgebruikers zich een compleet nieuwe werksituatie zouden moeten voorstellen en daar de Requirements voor zouden moeten beschrijven. Zij kunnen zich doorgaans geen beeld vormen van de toekomst, kunnen onvoldoende onderscheid maken tussen bestaande problemen of spanningen in de eigen organisatie en die uit het ICT proces voortkomen. Ook kunnen zij zich niet loslaten uit de eigen werkelijkheid, een werkelijkheid waarvan de praktische problemen niet eens meer herkenbaar zijn omdat met dit zo gewend is en daar met lapmiddelen 'tijdelijke' oplossingen voor hebben gemaakt. ICT-ers zeggen tegen deze gebruikers dat de Requirements probleemgericht moeten worden opgesteld en niet oplossingsgericht, want dat is de taak van de ICT-ers! Andersom verwijten ICT-ers gebruikers dat zij juist teveel de oplossing beschrijven en dat zij 'altijd' overvragen. Dat laatste is soms inderdaad waar, maar wat doe je in een organisatie waarvan je weet dat alles wat je vraagt met de helft wordt afgeknepen omdat dit nodig zou zijn of omdat hier onvoldoende geld voor beschikbaar zou zijn? Je overvraagt dan wat.

Bij product bedrijven, d.w.z. ICT bedrijven die standaard software pakketten aan de markt leveren, is het bepalen wat je wilt en het opstellen van Requirements nog een stuk lastiger dan voor de gemiddelde gebruiker in een doorsnee organisatie [zie referentie 9]. In dit soort organisaties moeten product managers Requirements opstellen voor onbekende gebruikers voor een toepassing die pas vele jaren na uitdenken en ontwikkeling op de markt komt en gebruikt zal gaan worden en in veel gevallen nog ettelijke jaren daarna zal blijven bestaan. Ga daar maar eens de Functional Requirements van beschrijven of beschrijf maar eens hoe je flexibiliteit of aanpasbaarheid voor nieuwe gebruikerswensen concretiseert! Je kent je gebruiker niet, weet niet wat hij met het pakket wil en kan gaan doen, laat staan wat zijn toekomstige wensen zouden zijn.

- Over het aanhouden van faseringen en te maken keuzes door gebruikers

De Requirements beperken zich ten onterechte vaak tot de middelen/processen/functies sfeer. Hoe hun werkprocessen veranderen wordt vaak als sluitpost van de begroting toegevoegd. Het is de vraag of gebruikers eigenlijk nog wel wat te kiezen hebben. Alles ligt immers toch al vast?! In de VOORKANT wordt een universeel model voor ontwikkeling neergezet, een model waarbij 'alle' disciplines 'alles' wat nodig is bedenken, ontwikkelen en goed op elkaar (blijven) afstemmen. Onderstaand model geeft aan dat het ook anders kan.



andersom 'HOE dit kan worden opgelost (Functioneel ontwerp), zou dit een adequaat antwoord kunnen geven als oplossingsrichting voor het probleem?' Dus opnieuw geldt hier: niet lineair denken en handelen maar iteratief tussen de verschillende betrokken disciplines. Merk op dat in VOORKANT het Technisch ontwerp wordt buitengesloten van de synthese. Bij overwegingen over Requirements wordt alleen functioneel oplossingsgericht gedacht. Hierbij wordt voorbij gegaan aan alternatieve technische oplossingsrichtingen.

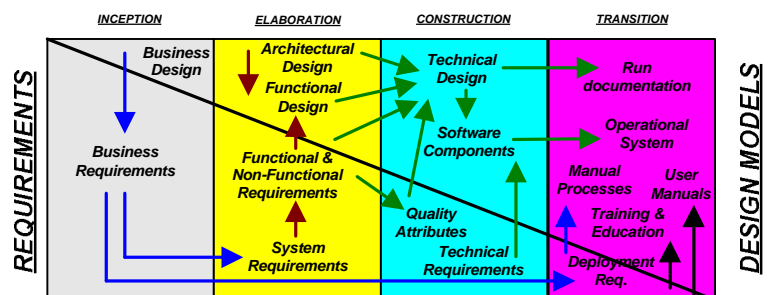
Gebruikers kunnen maar slecht tegen standaard aanpakken en faseringen waartoe ze vaak door ICT-ers toe worden gedwongen. De manier van waaruit standaard wordt gedacht en gewerkt vertegenwoordigt vaak een eigen wereld, zo'n eenzijdig referentiekader dat het voor de gebruikers moeilijk is daar iets uit de eigen wereld aan toe te voegen. Gebruikers ervaren ontwikkelmethoden als te star en te lineair. Telkens moeten deelbeslissingen worden genomen die blijkbaar alle in één keer juist zijn want veel ruimte voor latere bedenkingen is er niet. Ook veronderstellen ze gebruikers die (na enig nadenken) precies weten wat ze willen en dit exact definiëren. Open oordeelsvormingsprocessen zijn nodig om te leren van het voortschrijdend inzicht in het project en in het product in wording. Automatiseerders en nog sterker het management van automatiseerders, worden vaak erg zenuwachtig als er alleen maar wordt gepraat in het project, zoals ze dit noemen, en geen software wordt opgeleverd. Een goede voorbereiding is ook in dit geval (meer dan) het halve werk!

Wat in bovenstaand model opvalt dat voorbij de tweede verticale streep alleen maar 'uitwerking' plaatsvindt. Los van elkaar zullen Business- en ICT-specialisten hun deelsystemen bouwen die straks samengevoegd worden. Startconditie is dat het breed totaalbeeld van alle gewenste organisatie en ICT-functies duidelijk moet zijn, plus een nauwkeurig inzicht in hun raakvlakken. Dit volledig beeld moet primair aanwezig zijn in alle betrokkenen en pas in tweede instantie vastgelegd in ontwerp rapport, Requirements, proces- en organisatiebeschrijvingen en andere zaken. Het voorwerk, waarin de ruimte en beperkingen worden gecreëerd voor Systeemanalyse en Functioneel ontwerp, mag geen gedetailleerde systeem schetsen opleveren die de speelruimte en creativiteitodeloos verenigen. Er dient een synthese te bestaan tussen Systeemanalyse (probleemgericht) en Functioneel ontwerp (oplossingsgericht). Tijdens beide activiteiten moet vanuit beide gezichtspunten worden gekeken. WAT wordt gevraagd (Systeemanalyse) is dat wel oplosbaar en

• De oplossing die RUP biedt

De ontwikkelmethode RUP heeft bovenstaand dilemma aardig en elegant opgelost. De fasen Inception+ Elaboration van RUP vormen tezamen het onderzoek deel van het project (te vergelijken met de fasen tot aan de tweede rechter lijn in het bovengenoemd model), Construction+ Implementation vormen het oplever deel aan de rechterzijde van de tweede rechter lijn. In het onderzoek deel is er alle ruimte om Requirements te ontwikkelen en architectuur en andere modellen hiervoor te schetsen die de Requirements het beste zullen ondersteunen. Pas als dit voldoende is uitgekristalliseerd vervolgt het project met de uiteindelijke oplevering. Zonodig wordt een eerste increment volledig uitgewerkt en zelfs in productie genomen zodat gebruikers praktijkervaring kunnen opdoen en op basis daarvan hun Requirements voor toekomstige releases van het product herijken of herbevestigen.

Als ook de technische realisatiemogelijkheden in de synthese over het project wordt betrokken ontstaat een plaatje waarbij over alle stappen van ontwikkeling heen aandacht wordt geschonken aan beide facetten, aan zowel Requirements (eisen) als Design (oplossingen). Onderstaand model (vrij naar RUP) geeft dit weer.



Requirements en Design modellen wisselen elkaar af (perspectieven van waaruit naar de zaken wordt gekeken) en zijn in elke fase van toepassing.

- Over de grenzen van systemen en beperkingen die de organisatie oplegt

In de meeste gevallen kan worden verstaan met een korte opsomming van de grenzen waarbinnen het project moet bewegen, de zgn. 'nieten'. De 'nieten' betreffen de grenzen die aan het project en aan het systeem gesteld worden. Afbakening/scope, randvoorwaarden, uitgangspunten, beperkingen, passend in de Business Justification. Deze zaken zijn op te vatten als High Level Requirements, zaken waaraan in ieder geval aan tegemoet gekomen moet worden. De 'Nieten' zijn de Project Requirements. Deze staan boven aan de eerder genoemde Requirements piramide. Zo beperken de beschikbare tijd, geld en middelen de Business Requirements. Pas indien kan worden aangetoond dat deze Constraints zodanig knellen dat het op te leveren eindresultaat onder de maat zou zijn, kunnen eventuele wijzigingen hierin bespreekbaar worden gemaakt. Dit vergt herijking van de scope en afbakening van het project [zie referentie 2].

Samenvattend, de beschrijving van Requirements moet mee evalueren van:

- Probleem naar oplossingsgericht
- Grof naar fijn
- Onvolledig/onduidelijk naar volledig/duidelijk
- Breed naar smal
- Informeel naar formeel
- Gebruikerstaal naar automatiseringstaal

Dit vereist kennis en ervaring van het business domein, van het voortbrengingsproces, de informatiesystemen, de producten en diensten en de organisatie rondom dit alles. Verder zijn aspecten als de cultuur van de organisatie, macht, beloning en besluitvorming, afrekensystemen en haalbaarheid, mogelijkheden en onmogelijkheden ingrediënten die moeten worden geadresseerd. De volwassenheid van de organisatie en van de daarin uitgevoerde processen en van de mensen die er werken en het management daarvan zijn bepalend voor het goed kunnen toepassen van principes over Requirements and Acceptance Management. Methoden & Technieken en hulpmiddelen/tools ondersteunen dit maar het is de mens die bepaalt of ICT succesvol wordt ingepast.

Vaak blijken Requirements' gedaan in het voortraject meer suggesties te zijn, bij wijze van voorbeeld, dan dat dit 'harde' systeemeisen zijn. De Stakeholders die de Requirements indienen dienen deze zelf te onderbouwen (rechtvaardiging, toegevoegde waarde, noodzaak/nut) en te verdedigen tegen concurrerende Requirements van andere Stakeholders. In een eerder artikel over het Trade-off proces werd hier reeds op ingegaan [referentie 2].

• Conclusie

Er zijn verschillende methoden en technieken op de markt en even zo vele benaderingen die je vertellen hoe je het beste een Requirements proces kunt inrichten. Het is echter niet de methode die het 'm doet maar de mens die dit toepast en de organisatie die dit mogelijk maakt! Vooral als we in de BV Nederland in toenemende mate geconfronteerd worden met outsourcing en offshoring, wordt het belang van kraak heldere Requirements alleen maar groter. Het accent verschuift van het (ICT) werk uitvoeren naar besturen en controleren naar registreren. De Business organisatie voert dan de regie uit over meerdere partijen in de keten van Business Needs via Requirements en technische realisatie via Acceptatie naar eindresultaat en gebruik daarvan. In een volgend artikel wordt een Requirements & Acceptance proces geplaatst in een multi vendor / multi site / multi opdrachtgever / multi project omgeving en zal worden ingegaan hoe je zo'n proces op een volwassen wijze kunt inrichten en hoe je als organisatie vat blijft houden op de complexiteit van ICT (regiefunctie).

• Referenties

1. Serie Kwaliteit en Process Improvement - Requirements and Acceptance Management – deel 1, de Stakeholders (Martin Muller, SPIDER Koerier, juni 2005)
2. SPIDER 2005 – Serie Kwaliteit en Process Improvement - Requirements and Acceptance Management – deel 2, Het trade-off proces (Martin Muller, SPIDER Koerier, februari 2006)
3. DE VOORKANT van het automatiseren, Paukus B. Morssink en Aad Kranendonk, Stenfort Kroese 1987, ISBN 90 207 1565 8
4. CMMI Guidelines for Process Integration and Product Improvement, Mary Beth Chrissis, Mike Konrad, Sandy Shrum, Addison-Wesley 2003, ISBN 0421154967
5. Metaforen in Requirements engineering (Willem Visscher, Jonne Kats, Bas Terwijn, Jeroen Arnoldus en Daan van den Berg, Informatie/oktober 2005)
6. Start uw eigen weg naar (IT)-volwassenheid De Starterkit voor procesverbetering bij kleine(re) IT-eenheden of organisaties, 2002 SPIDER-werkgroep "SPI in Kleine Organisaties (www.st-spider.nl)
7. Requirements-Driven Management: A Planning Language (Tom Gilb, juni 1997 (http://www.stsc.hill.af.mil/crosstalk/1997/06/requirements.asp)
8. Raamwerk Kwaliteit (Martin Muller, SPIDER Koerier, mei 2005)
9. A Reference Framework for Software Product Management, Inge van de Weerd, Sjaak Brinkkemper, Richard Nieuwenhuis, Johan Versendaal, Lex Bijlsma, 2006 Universiteit Utrecht

Martin Muller

E-mail: Martin.Muller@LogicaCMG.com

In het najaar van 2007 organiseren Bits&Chips en het Embedded Systems Institute de conferentie:

Bits&Chips 2007 Embedded Systemen

Het lezingenprogramma zal bestaan uit parallelle tracks met bijdragen uit de academische wereld en de industrie. Het thema is 'prestatie en betrouwbaarheid' (performance and reliability) van embedded systemen.

Bits&Chips en het Embedded Systems Institute nodigen onderzoekers en professionals uit de industrie uit om voorstellen of artikelen in te sturen rondom dit onderwerp. U kunt een abstract van maximaal 400 woorden of een academisch paper indienen. Voor beide categorieën, academisch en industrie, buigen afzonderlijke commissies zich over de aangeleverde stukken.

De academische bijdragen worden beoordeeld door:

Ed Brinksma (ESI) . Henk Corporaal (TU Eindhoven) . Arjan van Gemund (TU Delft) . Boudewijn Haverkort (Universiteit Twente) . Nieke Roos (Bits&Chips)

De industriële bijdragen worden beoordeeld door:
Frans Beenker (ESI) . Michiel van der Korst (Thales) . René Krikhaar (ICT NoviQ) . René Raaijmakers (Bits&Chips)

In beide commissies nemen zitting:

Teade Punter (ESI) . Eric van Utteren (Progress/Philips)

Inzendingen moeten bevatten:

- Titel
- Naam en functie auteur
- Portretfoto auteur
- Correspondentieadres

Abstracts/papers dienen het LNCS-formaat te hebben (zie www.springer.com/lncs -> For Authors) en kunt u opsturen naar Teade Punter (teade.punter@esi.nl) en Nieke Roos (nieke@bits-chips.nl).

Geaccepteerde abstracts kunnen alsnog worden omgezet in een paper.

Goedgekeurde academische artikelen worden gepubliceerd in de handelingen van de conferentie. Geaccepteerde papers uit de industrie kunnen in overleg worden opgenomen in de proceedings of als artikel in Bits&Chips.

Belangrijke deadlines:

12 maart 2007 - laatste inzenddatum paper

30 april 2007 - uitslag over acceptatie

25 juni 2007 - laatste inzenddatum zetklaar paper

Een verslag van Ben Linders over zijn verblijf van 5 weken bij het SEI in het kader van een affiliate assignment.

• Onderzoek naar kwaliteitsfactoren

Bij Ericsson ben ik al een aantal jaren bezig met het modelleren en meten aan kwaliteit, en als kwaliteitsman in hart en nieren wilde ik een keer dieper in de materie gaan. Ik besloot dat onderzoek naar de oorzaken van kwaliteit mijn volgende stap zou zijn. Maar, waar zou ik dat soort onderzoek kunnen doen? Bij het Software Engineering Institute (SEI) bleken er mogelijkheden te zijn, als een affiliate van de Software Engineering Measurement & Analysis group. Begin dit jaar heb ik een onderzoeksopdracht geformuleerd, en ben ik "long distance" vanuit Nederland de samenwerking begonnen met het SEI. In oktober ben ik, als sabattical, 5 weken "on-site" geweest in Pittsburgh om full time te werken aan mijn onderzoek. Hoe ziet zoiets er dan uit? Hieronder een aantal pagina's uit mijn dagboek, die een indruk geven van mijn ervaringen.

25 september: Measuring for Performance Driven Improvement

De eerste week van mijn on-site periode volg ik de cursus Measuring for Performance Driven Improvement. Deze cursus laat zien hoe je metingen kunt gebruiken om software ontwikkeling te verbeteren, gebaseerd op Six Sigma DMAIC, een methodiek die zwaar gebruik maakt van statistiek, en de nadruk legt op vooraf definiëren en achteraf bewijzen van de business case. Da's niet niks, als je nagaat dat alleen al het goed meten aan organisaties iets is wat al snel een half jaar kost voordat je resultaten hebt. Met een Six Sigma project, wat uitgevoerd wordt door een z.g. black belt, zou je in dat zelfde half jaar ook nog meetbaar iets verbeterd moeten hebben, wat dus bijdraagt aan het eindresultaat van het bedrijf. Ga er maar aan staan! En dat is dus precies wat ik ga doen. Mbv Six Sigma laten zien dat we in staat zijn om onze kwaliteit dusdanig te verbeteren dat onze klanten daar iets aan hebben. Gelukkig heb ik iets meer dan een half jaar, aangezien ik met dit onderwerp al enkele jaren bezig ben, en al een meetmodel heb wat gebruikt wordt door projecten. Anderzijds krijgt een black belt minimaal 5 weken training, en ik zal het met 1 week en personal coaching moeten doen. Dus het blijft een uitdaging.

4 oktober: In God we trust, all others bring data

Vandaag ga ik toepassen wat ik vorige week geleerd heb: Data analyse! Ik was gewaarschuwd, als je dat gaat doen dan zal blijken dat je data vaak vervuild is. En inderdaad, ook mijn kwaliteitsdata van alle projecten klopt niet overal. Met de analyse zie je meteen waar er rare dingen zijn. Gelukkig heb ik alle files van Ericsson bij me, dus ik kan als ik iets gekk zie meteen inzoomen op het project en kijken wat er aan de hand is, en het ook repareren. Intensief proces, maar wel nodig als je betrouwbare getallen wilt hebben. Daarnaast heb ik ook nog planning data van een recent project. Deze data moet ik nog ingeven en linken naar de gevonden fouten, zodat ik kan zien hoeveel uur het vinden van

een fout kost. Maar ook dit klopt niet helemaal, mensen zijn niet altijd even consequent in hun urenrapportage. De conclusies uit de analyse gaan in een email terug naar de projectmanager in NL, met de check om geboekte uren aan te passen. Ook de Excel files gaan terug, maar dan rechtstreeks in de project directory. Met de laptop kan ik overal bij, werkt perfect! Het is iets trager en soms een beetje omslachtiger (allemaal web based), maar qua mogelijkheden kan het allemaal. 2 offices in 1, techniek staat voor niets!

6 oktober: Kwaliteit zit overal

Het is een intensief 2e weekje geweest. Inmiddels heb ik een geordende opsomming van allerlei factoren die kwaliteit beïnvloeden gemaakt, met een voorstel hoe ze te meten zijn. Dit gebaseerd op eerdere literatuurstudies, conferenties, mijn ervaring en kennis, en de kennis en ervaring van het SEI. Het document gaat op de mail naar mijn mentor, zodat we het volgende week kunnen reviewen. Dus ik heb wel een weekendje vrij verdiend. Maar eerst nog in mijn appartement testen of ik via internet overal bij kan. Ik kan met VPN en de Virtual Manager op mijn SEI PC, en dus bij de SEI network drives. Verder werkte het SEI intranet al, het CMU intranet met mijn email, en de BSCW met de files van mijn studie, en alle Ericsson zaken dus ook daar kan ik de hele wereld aan. Inclusief ook mijn privé webmail, MSN, Skype, VoiPbusters, elektronisch bankieren, Credit cards en whatever. Alles op een laptopje met high speed cable internet!

11 oktober: Pittsburgh SPIN

Vandaag is er een bijeenkomst van de Pittsburgh SPIN (Software Process Improvement Network). Als voorzitter van de NL SPIN zit ik in een SEI netwerk, met alle andere SPINs. De SPIN komt hier maandelijks bijeen, altijd op de 2e woensdag. Locatie varieert, ergens in en rondom Pittsburgh. 1 spreker, geen koffie of broodjes. De spreker vandaag is Mike Phillips van het SEI, een oude bekende die ik op kantoor nog niet getroffen had, dus dan zie je elkaar hier, toch? Opkomst 10 personen, varieert tussen de 10 en 25. Ze hebben een paar honderd leden op de mailinglist. Is toch heel anders dan SPIDER, waar we meer dan 1000 leden hebben, op de plenaire sessies rond de 50 deelnemers zitten, en 3-4 sprekers per sessie. We doen het nog niet zo slecht in Nederland...

13 oktober: Tussenstand

Week 3 zit er bijna op, dus tijd voor een tussenstand. De Six Sigma DMAIC aanpak zit er aardig in, inclusief eerste oefeningen met minitab voor data analyse. De Ericsson data is al een eind gereed voor analyse, hoewel daar er wel wat werk verzet moet worden in NL om alle foutinformatie up to date te krijgen. Maar daar wordt hard aan gewerkt, en ik mail en bel regelmatig met de collega's om mijn input te krijgen. Er staat een overzicht van de kwaliteitsfactoren, inclusief meetdefinities. En ik heb een goed beeld hoe Ericsson staat m.b.t. defect detection en prevention, en wat er "te koop" is in de software (research) wereld. Tevens nog een presentatie gegeven van

mijn Project Defect Model voor de scientists bij het SEI. Contacten met diverse SEI scientists, en contact gehouden met het Ericsson thuisfront. Een "Technical Diary" bijgehouden, waarin staat wat ik allemaal gedaan en bekeken heb, en eventuele conclusies. Niet slecht voor 3 weken on-site! Ok, dit kon ook alleen omdat ik al heel veel had voorbereid. Maar toch, snelheid zit er in, en samen met het SEI levert dat resultaten.

19 oktober: Reference group meeting

Belangrijke dag vandaag. Ik heb een phone conference met mijn "Reference Group", dat zijn de unit managers bij Ericsson die de resultaten van mijn research project nodig hebben. De kunst is om ze betrokken te houden, zeker als je een dikke maand in het buitenland zit en dus niet zichtbaar bent. Daarnaast heb ik van hun een goedkeuring nodig voor het vervolgplan met een survey die ik wil doen binnen Ericsson. Aan het begin is het een gezellig gekakel, met wat grappen over en weer over zon in Nederland en sneeuw in Pittsburgh, maar we hebben maar ½ uur dus, back to business! Ik heb 2 alternatieven tav de planning voor mijn opdracht, en ze kiezen degene die mijn voorkeur heeft: uitgebreid onderzoek en daarna met mijn model de verbetergebieden bepalen. Yesss, dan kan ik weer verder, sterker nog, zal er even aan moeten gaan trekken om die planning te halen. Maar de reacties van mijn reference group geven mij energie om door te gaan.

23 oktober: Quality factors survey

Met een senior scientist van het SEI ga ik aan de slag om een survey op te zetten over kwaliteitsfactoren. We zijn eerst de nodige tijd kwijt om de tool op mijn laptop aan de praat te krijgen, en dan kunnen we aan de slag met de vragen. Dat komt best nauw, de manier waarop je een vraag stelt kan tot heel verschillende antwoorden leiden! De SEI scientist heeft "wat" ervaring met surveys: zijn eerste survey deed hij in 1962, om voorspellingen te doen m.b.t. de verkiezingen. I wasn't even born then! Yep, en hij had destijds op de resultaten ook maar meteen wat statistiek losgelaten, want dat wilde hij graag leren, zijn achtergrond was immers in de sociologie en psychologie. Later heeft ook nog e.e.a. bijgeleerd over computers, maar ach, zijn kennis van mensen komt toch nog regelmatig van pas. Zo ongeveer dagelijks. Met zo iemand kan ik wel iets, en vraag na vraag krijgt mijn survey body. Gaat niet snel, maar quality first!

27 oktober: Wegwezen!

Het zit erop, wat betreft het on-site deel. Vandaag de laatste spullen ingepakt, alle boeken, artikelen en verdere papierwerk gaan via een koerier naar Nederland. Dat scheelt flink in mijn bagage. Ik heb een geweldige tijd hier gehad, heel veel geleerd. En alleen al dat je een onderwerp helemaal kunt uitdiepen, en daar met vakgenoten doorheen kunt gaan is een geweldige ervaring. De Six Sigma DMAIC methodiek past perfect op de behoefte aan betere business cases en meetbare resultaten. Nu verder in Nederland met mijn onderzoek bij Ericsson. Mijn affiliate assignment loopt tot 31

augustus 2007, maar tussentijds ga ik al wel publiceren via het SEI. Dus, wordt vervolgd!

Mocht je meer willen weten over affiliates bij het SEI, kijk dan op:

<http://www.sei.cmu.edu/collaborating/affiliates/>

Ben Linders

Ericsson Telecommunicatie B.V. en Affiliate Software Engineering Measurement & Analysis bij het SEI.

n Verslag workshop roadmaps

Op 15 november jongstleden was de workshop voor het uitwerken van de CMMI roadmaps. De CMMI roadmaps zijn bedoeld als houvast voor organisaties die de continue representatie van CMMI willen gebruiken, maar moeite hebben een eerste keuze uit de 23 procesgebieden te maken. Afhankelijk van de doelstellingen van het verbetertraject, problemen die een organisatie op wil lossen en het type organisatie kiest een organisatie een bepaalde roadmap. Een roadmap bestaat uit een beperkt maar samenhangende set procesgebieden en mogelijke vervolgstappen. Meer informatie over het concept roadmap vindt u op de website www.sysqa.nl, onder publicaties staat een artikel en een presentatie over de roadmaps.

Na de lancering van het idee van roadmaps hebben een aantal Spider-leden, te weten, André Heijstek, Ben Linders, Rini van Solingen en ondergetekende, het idee opgevat om in samenwerking met de leden van Spider het concept roadmaps verder uit te werken met als doel er een technical note van het Software Engineering Institute van te maken. Als dat lukt wordt dit dus een onderdeel van de officiële CMMI productsuite, die vanuit de Nederlandse SPI-community is voortgekomen.

Bij de workshop waren een kleine 40 deelnemers aanwezig. Na een korte presentatie zijn in vijf groepen de tot op dat moment onderkende roadmaps uitgewerkt, dit zijn Project, Product, Product Integratie, Proces en Performance. Per roadmap hebben de groepjes een aantal vragen beantwoord zoals welke doelstellingen de roadmap heeft, welke problemen opgelost worden met de roadmap, welke procesgebieden bij de roadmap horen en welke mogelijke vervolgstappen na de roadmap genomen kunnen worden. Binnen de groepen werd flink gediscussieerd en kreeg iedereen de kans zijn bijdrage te leveren. De uitkomsten zijn vervolgens aan elkaar gepresenteerd. Deze uitkomsten zijn in presentaties vervat, de presentaties zijn voor de deelnemers aan de workshop te downloaden vanaf de website van Spider (www.st-spider.nl).

Uiteindelijk leverde deze opzet een succesvolle avond op. Naast de doelstelling om de roadmaps uit te werken was de doelstelling om er een interactieve avond van te maken waar ruimte was voor de ideeën van alle deelnemers en ervaringen gedeeld konden worden. Gezien de positieve reactie kunnen we stellen dat ook deze doelstellingen gehaald zijn.

Nu zijn de initiatiefnemers weer aan zet. Zij werken op dit moment de uitkomsten uit in de vorm van een SEI-technical note (valt nog niet mee, een rapport in

het Engels schrijven). Zodra er tussen de initiatiefnemers overeenstemming is over de inhoud wordt het conceptrapport voorgelegd aan de deelnemers van de workshop. Die kunnen het rapport dan reviewen en hun commentaar leveren. Nadat het commentaar beoordeeld en verwerkt is wordt het rapport aangeboden aan het SEI. Ik ben wat huiverig een geplande datum voor de review te roepen. Ik heb een leuke baan, maar ja, die klanten kosten zoveel tijd. Daarnaast is het moeilijk te voorspellen hoeveel tijd het kost om met de initiatiefnemers op één lijn te komen.

Bij deze wil ik nogmaals de sponsors van de avond, LogicaCMG en SYSQA, bedanken. Daarnaast een dankwoord in de richting van Spider, zij maken het iedere keer weer mogelijk dit soort bijeenkomsten te organiseren en als vakbroeders met elkaar over ons vak te praten en van elkaar te leren.

Jan Jaap Cannegieter
Adjunct Directeur SYSQA B.V.

n Verslag Q sessie

Op 6 december jl. werd de middag/avond gereserveerd voor een sessie georganiseerd door de "Society for Quality Professionals in ICT", kortweg Q Society genoemd. In dit artikel worden kort wat opvallende zaken toegelicht. De opzet was afwijkend van die van de vorige Q sessie van 11 mei van dit jaar. De locatie was opnieuw het NBC in Nieuwegein. Er waren vooraf 4 stellingen geformuleerd rondom het thema: "Kwaliteit in ICT, DAT kan IK een stuk beter!" Deze titel werd gekozen om een discussie los te maken rondom kwaliteit door practitioners in kwaliteit maar ook afnemers van kwaliteit(producten in ICT) te stimuleren om actief met elkaar over kwaliteit issues aan de slag te gaan.

Stellingen

1. De introductie van kwaliteit werkt juist contraproductief
2. Het kwaliteitsprobleem is onvoldoende zichtbaar
3. De oorzaken van onvoldoende kwaliteit zijn niet bekend
4. Kwaliteit is puur een vraagstuk voor seniormanagement (stelling behorende bij de discussie "Leggen we deze vraagstukken wel aan de juiste personen voor?")

Presentatie vooraf

Martin Muller trapte de workshops af met een verhaal waarin emotie over het gebrek aan kwaliteit sterk doorklonk! Het is toch te gek voor woorden dat projecten nog steeds niet opleveren wat ervan wordt verwacht, dat men gewoon raakt aan fouten en workarounds rondom fouten en dat ondanks vele methoden,

technieken, hulpmiddelen en certificering van mensen het lijkt of er nauwelijks iets veranderd is ten opzichte van 10 jaar geleden. Martin gaf daarnaast aan dat er meer dan genoeg uitdagingen liggen voor de BV Nederland, bijvoorbeeld op het gebied van het aansturen van uitbesteed werk aan verwegistan. Als extra bonusvraag daagde Martin zijn publiek uit om het aantal fouten in zijn eigen presentatie te tellen! Helaas zijn we daar door het best wel volle programma, niet meer aan toegekomen. Voor degenen die erbij waren, er zaten 18 fouten in de presentatie... Waar, waarom en hoe geteld ga ik een ander keer op in.

Na de openingspresentatie van Martin leidde Jan Dobbels ons via "Auw! Dat doet pijn" in zijn wereld van productontwikkeling en architectuur bij Siemens/VDO. Jan had uitdagende stellingen uitgewerkt om het publiek te prikkelen en ze tot nadenken aan te zetten. Nou dat is gelukt hoor! Een aantal mensen was het dus niet met bepaalde uitspraken eens. Prima vinden wij, dat brengt juist een discussie op gang waar we allemaal van kunnen leren. Een leuke stelling was de volgende: "Kwaliteitsverbetering komt niet voorbij het niveau van de quick win

- Budgetten worden jaarlijks verstrekt
- Resultaat moet binnen die periode plaats vinden
- Verbetering wordt echter niet echt gemeten
- Quick wins 'helpen' dan om nieuw budget in de wacht te slepen

zorgden aan het eind van de workshops voor de terugkoppeling in de plenaire groep. In totaliteit waren er 8 groepen van maximaal 10 personen. De stelling stond centraal bovenaan het metaplan bord vermeld. Na discussie werden reacties kernachtig verwoord op het metaplan bord geplaatst. Per stelling/groep was er 30 minuten discussie. Daarna 'changez', de deelnemers gingen naar een andere ruimte waar een andere moderator al klaar stond om de volgende stelling te behandelen.

Afsluiting

De dag werd afgesloten met een aangeklede borrel en gelegenheid tot netwerken. De sfeer tijdens de workshops en ook daarna was ongedwongen en zeer plezierig. Met 75 mensen die aanwezig waren kan je zeggen dat dit een geslaagde dag was! Met de vertegenwoordigers van de 8 organiserende bedrijven werd tot laat in de avond nog doorgediscussieerd over de organisatie van het voorjaar event 2007. Wij mikken op de tweede week van mei. In de eerste Koerier van 2007 zullen wij verslag doen van de opzet van het inmiddels derde Q event.

De die dag gemaakte foto's zullen op de website worden geplaatst evenals de presentaties en een verslag van de dag. Onderstaande foto geeft een impressie van de locatie. De organisatoren van de dag werden bij afsluiting op de foto gezet. Helaas ontbrak één van hen. Dat corrigeren we een andere keer wel.



Een andere uitspraak van Jan was: "Kwaliteit maakt zich eigenaar van de kwaliteitsproblemen van een organisatie". Kortom genoeg voer tot nadenken en discussie.

De opzet workshops o.l.v. moderatoren

De deelnemers werden per stelling gegroepeerd rondom een moderator. De moderatoren begeleidden de discussies, legden de uitkomst van de discussies vast en

Ik sluit af om alle deelnemers te bedanken voor hun inzet op die dag en namens de Q Society wensen wij U alvast prettige feestdagen en een voorspoedig improved 2007.

Martin Muller

Voorzitter Q Society, bestuurslid SPIDER

n De werkgroepen

n Werkgroep SPI in kleine organisaties

De Werkgroep "SPI in kleine organisaties" houdt zich vooral bezig met alle aspecten van procesverbetering in omgevingen tot zo'n 50 mensen. Dat kunnen natuurlijk ook afdelingen van grotere organisaties zijn. De werkgroep houdt een lijst van onderwerpen bij, en vult die minstens eenmaal per jaar aan. Altijd geldt dat de onderwerpen voortspruiten uit de praktijk van de leden. Negen van de tien keer dan ook kun je als lid een nieuw verworven inzicht de volgende dag al in je eigen werk toepassen. De groep is een werkgroep, met andere woorden: de leden halen en brengen kennis. Iedereen toetst zijn eigen praktijk aan hetgeen besproken wordt; in de werkgroep wordt ook tijd besteed aan het uitwisselen van ervaringen en inzichten. De groep biedt al met al een uitstekende kans om over het vak "SPI" te praten buiten de directe werkomgeving.

Zie voor meer informatie de website van de werkgroep, te bereiken via www.st-SPIder.nl. Wilt u een bijeenkomst van de werkgroep "SPI in kleine organisaties" bijwonen, neem dan contact op met: Herman Rave (voorzitter), tel: 06-53231662, herman@raveonline.nl, of Tjeu Naus, tel: 0495-633221, Tjeu.Naus@nbg-industrial.nl.

n Werkgroep Metrieken

De werkgroep houdt zich in ruime zin bezig met metingen aan software projecten, software ontwikkelprocessen en software producten onder het motto "Quality without numbers is just talk". Daarbij gaat het zowel om de definitie, invoering en analyse van metrieken, als om de resultaten van die metingen (benchmarking).

Contactpersoon: Robert van Lieshout

Tel: 06-23921989

E-mail: r.v.lieshout@simbon.nl

n Werkgroep SPI Invoeringsstrategieën

De SPIder Werkgroep Invoeringsstrategieën richt zich in ruime zin op alle facetten die te maken hebben met het invoeren van nieuwe werkwijzen. Belangrijke aspecten zijn daarbij het delen van ervaringen en meningen, het bieden van een klankbord voor het bespreken van ideeën en problemen en het volgen van nieuwe ontwikkelingen op het gebied van SPI. Het principe "halen én brengen" is één van de belangrijkste kenmerken van onze werkgroep.

Contactpersoon: André Heijstek

Tel: 0182-689321 / 06-48476451, e-mail:

Andre.Heijstek@ImprovementFocus.com

n Werkgroep Integrale SPI strategieën

De werkgroep richt zich op de toegevoegde waarde (return on investment) van procesverbetering en kwaliteitsmodellen. Binnen de werkgroep zijn daarvoor een aantal kwaliteitsmodellen geselecteerd, die nader worden uitgewerkt. Per

kwaliteitsmodel wordt een datasheet gemaakt met daarin opgenomen:

- een korte beschrijving;
- toepassingsgebied (aandachtsgebied, zoals informatiemanagement, beheer, systeemontwikkeling, projectmanagement);
- geografische positie (waar toegepast);
- soort model (verbetermodel, referentiemodel, beheersmodel, procesmodel, etc.);
- eigenaar;
- beschikbaar sinds;
- relaties met andere modellen.

Het doel van de werkgroep is om, op basis van deze korte modelbeschrijvingen, te komen tot een "consumentenbondoverzicht" waarbij het gebruik en samenhang van de diverse modellen in beschreven is. Het motto van de werkgroep is: "Modelpatronen: Welke verbetering levert dit op voor de klant?"

Momenteel zijn de volgende modellen onderzocht en beschreven: CMM(i), INK/EFQM, ISO, RUP, ISPL, Six Sigma, Itil, Cobit, IT Service CMM, SPICE, ASL, DSDM, Prince2, IPW, Tmap, TPI, TMM, MOF/MSF, BISML, TickIt.

Contactpersoon: H.J. Steures

Ben je geïnteresseerd? Neem dan contact op met H.J.Steures@wanadoo.nl

n Nieuwsberichten & evenementenkalender

De evenementenkalender bevat een overzicht van internationale conferenties op het gebied van SPI, metrieke en softwareproductkwaliteit. Daarnaast zijn de activiteiten van SPIDER opgenomen.

Ook nationale evenementen op het gebied van softwareproduct- en procesverbetering kunnen in deze evenementenkalender worden opgenomen. Via de SPIDER Koerier kan een organisator van SPI gerelateerde evenementen een selecte groep van geïnteresseerden bereiken. Voor commerciële evenementen zoals conferenties, workshops, lezingen en andersoortige bijeenkomsten vraagt de redactie een kleine bijdrage in de kosten.

Ⓢ = SPIDER event

Ü = korting voor SPIDER donateurs

2007

- | | | |
|-----------|---|---|
| 6-9 jan | WICSA conferentie, Mumbai, India
http://www.wicsa.net | |
| Ⓢ Feb | SPIDER plenaire sessie | |
| 21 mrt | European Conference Software Maintenance and Reengineering
http://www.cs.vu.nl/csmr2007/ | |
| 23 mrt | Verification & Validation of Software Systems (VVSS)
http://www.laguso.com/VVS2007/ | |
| 26-29 mrt | SEPG conferentie, Austin, Texas
http://www.sei.cmu.edu/sepg/2007/ | |
| 17 apr | ITSMF voorjaarscongres
http://www.best-practices-in-itsm.nl/ | |
| 8 mei | PMI Parade: Progressie in projectmanagement
http://www.pmi-nl.nl/ | |
| Ⓢ Mei | Voorjaarsconferentie Kwaliteit van de Society for Quality Professionals in ICT | Ü |
| 11 jun | ESEPG Conference
http://www.espi.org | Ü |
| Ⓢ Okt | SPIDER Conferentie 2007 | Ü |

n Deelname in SPIDER

Indien u actief wilt participeren in SPIDER en de Koerier in de toekomst wilt ontvangen, kunt u zich aanmelden als deelnemer in SPIDER bij:

Secretariaat Stichting SPIDER

p/a Cantrijn Secretariaten
Postbus 2047, 4200 BA GORINCHEM
Tel: 0183 - 62 00 66, fax: 0183 - 62 16 01
E-mail: info@st-SPIDER.nl

Aanmelding kan ook via het aanmeldingsformulier op de website van SPIDER: www.st-SPIDER.nl.

n Colofon

De SPIDER redactie bestaat uit:
Cees Michiels en Cantrijn Secretariaten.

Voor reacties en vragen m.b.t. de **SPIDER Koerier** kunt u zich wenden tot:

Redactie SPIDER Koerier

E-mail: koerier@st-SPIDER.nl

Indien u in de toekomst een herinneringsbericht wilt ontvangen over de datum van kopijsluiting, stuur dan een e-mail "opname SPIDER copylijst" naar Koerier@st-SPIDER.nl.

Informatie over SPIDER is te vinden op de website: www.st-SPIDER.nl.

Voor reacties en bijdragen op de **SPIDER website** kunt u zich richten tot:

Redactie SPIDER web, Niels Malotaux

E-mail: niels@malotaux.nl

Deze koerier kwam tot stand met medewerking van:

- [ITIB](http://www.itib.net) (www.itib.net)

- Cantrijn Secretariaten

All engineering is characterized by the engineer's dissatisfaction with the achievement of just a solution. Engineering seeks the best solution in established terms, within recognised limitations, and making compromises required by working in the real world.

[E. Yourdon, L. Constantine]